securityandtrust.lu

Example-Dependent Cost-Sensitive Logistic Regression for Credit Scoring

December 5, 2014

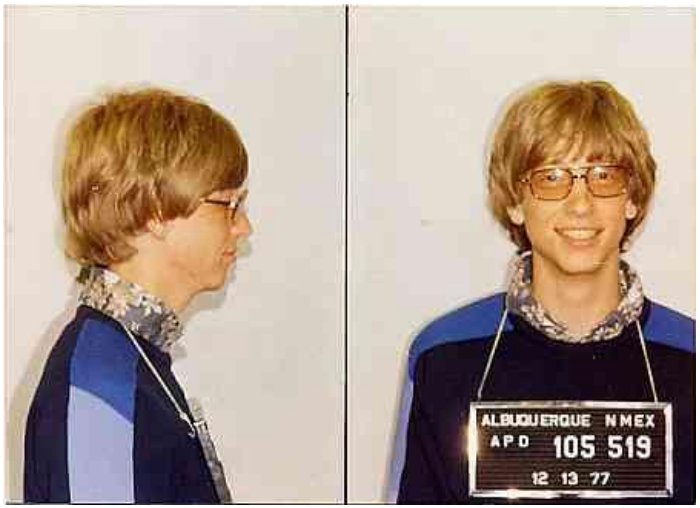
Alejandro Correa Bahnsen

with

Djamila Aouada, SnT
Björn Ottersten, SnT

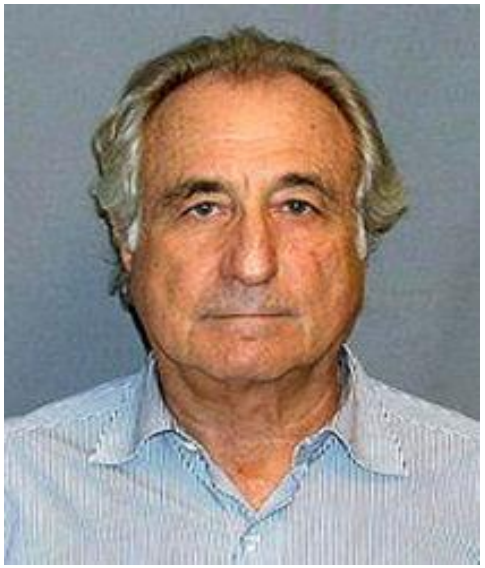
Credit Scoring - Example

- Just fund a bank



- Just quit college

Credit Scoring - Example



- Biggest Ponzi scheme

- Now Billionaire

Credit Scoring

- Mitigate the impact of **credit risk** and make more objective and accurate decisions
- Estimate the **risk of a customer defaulting** his contracted financial obligation if a loan is granted, based on past experiences
- Different ML methods are used in practice, and in the literature: **logistic regression**, neural networks, discriminant analysis, genetic programing, decision trees, among others

Credit Scoring

- Evaluation of credit score models
  - Brier score
  - AUC
  - KS
  - F1-Score
  - Misclassification

- Nevertheless, none of these measures takes into account the **business and economic realities** that take place in credit scoring. Different costs that the financial institution has incur to acquire customers, or the **expected profit** due to a particular client, are not incorporated in the evaluation of the different models

Credit Scoring

- **Financial evaluation** of credit score models

	Actual Positive $y_i = 1$	Actual Negative $y_i = 0$
Predicted Positive $c_i = 1$	$C_{TP_i} = 0$	$C_{FP_i} = r_i + C^a_{FP}$
Predicted Negative $c_i = 0$	$C_{FN_i} = Cl_i \cdot L_{gd}$	$C_{TN_i} = 0$

- Correct classification costs are assumed to be 0
- C_FN = **losses** if customer i defaults
- Cl_i is the **credit line** of customer i
- Lgd is the **loss given default**. Percentage of loss over the total credit line when the customer defaulted

Credit Scoring

- **Financial evaluation** of credit score models
- $C_{FP_i} = r_i + C^a_{FP}$
- $C^a_{FP} = -\overline{r} \cdot \pi_0 + \overline{Cl} \cdot L_{gd} \cdot \pi_1$
- **loss in profit** by rejecting what would have been a good customer
- assumption that the financial institution will **not keep the money of the declined customer idle**, but instead it will give a loan to an alternative customer
- Whom as an average customer has default probability equal to the prior default probability π_1

Credit Scoring

securityandtrust.lu

- **Financial evaluation** of credit score models

	Actual Positive $y_i = 1$	Actual Negative $y_i = 0$
Predicted Positive $c_i = 1$	$C_{TP_i} = 0$	$C_{FP_i} = r_i + C^a_{FP}$
Predicted Negative $c_i = 0$	$C_{FN_i} = Cl_i \cdot L_{gd}$	$C_{TN_i} = 0$

$$Cost(f(S)) = \sum_{i=1}^{N} \Big(y_i(c_i C_{TP_i} + (1-c_i)C_{FN_i}) + (1-y_i)(c_i C_{FP_i} + (1-c_i)C_{TN_i}) \Big).$$

$$Savings(f(S)) = \frac{Cost(f(S)) - Cost_l(S)}{Cost_l(S)}.$$

UNIVERSITÉ DU LUXEMBOURG

Experiments

- Two **publicly** available datasets
  - Kaggle Credit dataset
  - PAKDD Credit dataset

- Contains information regarding **customers income and debt** from which the credit limit can be inferred, see appendix.

Table 2. Model parameters

Parameter	Kaggle Credit	PAKDD Credit
Interest rate (int_r)	4.79%	63.0%
Cost of funds (int_{cf})	2.94%	16.5%
Term (n) in months	24	24
Loss given default (lgd)	75%	75%

Experiments

- Using Decision Trees (**DT**), Logistic Regression (**LR**) and Random Forest (**RF**) to estimate the probabilities

- Databases partitioned in training (**t**), validation and testing
- Each of them contain 50%, 25% and 25% of the total examples, respectively

- Under-sampled (**u**) dataset

- SMOTE - Synthetic Minority Over-sampling Technique (**s**)

Experiments

securityandtrust.lu

- **Savings** of the DT, LR and RF algorithms

set	Algorithm	Kaggle Credit dataset	PAKDD Credit dataset
t	DT	19.88	-8.36
	LR	2.87	0.38
	RF	15.83	3.25
u	DT	34.49	-20.0
	LR	43.63	15.81
	RF	49.63	9.65
s	DT	3.46	-4.56
	LR	40.12	15.43
	RF	3.01	0.0

UNIVERSITÉ DU LUXEMBOURG

Cost-Sensitive Classification

securityandtrust.lu

- Changing **class distribution**
  - Cost Proportionate Rejection Sampling
  - Cost Proportionate Over Sampling
- **Direct Cost**
  - Bayes Minimum Risk
- **Modifying** a learning algorithm
  - Cost-Sensitive Logistic Regression

Experiments

securityandtrust.lu

Cost-Sensitive Sampling

set	Algorithm	Kaggle Credit dataset	PAKDD Credit dataset
t	DT	19.88	-8.36
	LR	2.87	0.38
	RF	15.83	3.25
u	DT	34.49	-20.0
	LR	43.63	15.81
	RF	49.63	9.65
s	DT	3.46	-4.56
	LR	40.12	15.43
	RF	3.01	0.0
r	DT	33.57	7.59
	LR	33.14	22.97
	RF	**50.01**	**30.1**
o	DT	19.6	8.95
	LR	33.56	23.03
	RF	21.69	23.26

Bayes minimum risk

set	Algorithm	Kaggle Credit dataset	PAKDD Credit dataset
t	$DT - BMR$	13.47	27.22
	$LR - BMR$	29.14	29.38
	$RF - BMR$	49.39	30.11
u	$DT - BMR$	34.58	26.49
	$LR - BMR$	45.25	29.6
	$RF - BMR$	51.47	31.14
s	$DT - BMR$	-0.54	26.84
	$LR - BMR$	43.26	29.63
	$RF - BMR$	43.11	26.75
r	$DT - BMR$	33.58	25.99
	$LR - BMR$	35.6	29.98
	$RF - BMR$	50.57	28.11
o	$DT - BMR$	11.77	27.12
	$LR - BMR$	43.09	29.53
	$RF - BMR$	49.38	28.03

uni.lu UNIVERSITÉ DU LUXEMBOURG

Cost-Sensitive - Logistic Regression

- Logistic Regression Model

$$\hat{p}_i = P(y = 1|X_i) = h_\theta(X_i) = g\left(\sum_{j=1}^{k} \theta^j x_i^j\right)$$

- **Cost Function**

$$J_i(\theta) = -y_i \log(h_\theta(X_i)) - (1 - y_i) \log(1 - h_\theta(X_i))$$

- **Cost Analysis**

$$J_i(\theta) \approx \begin{cases} 0 & \text{if} \quad y_i \approx h_\theta(X_i) \\ \text{inf} & \text{if} \quad y_i \approx (1 - h_\theta(X_i)) \end{cases}$$

$$\Rightarrow \quad \begin{aligned} C_{TP_i} &= C_{TN_i} \approx 0 \\ C_{FP_i} &= C_{FN_i} \approx \text{inf} \end{aligned}$$

Cost-Sensitive - Logistic Regression

- **Actual** Costs

$$J_i^c(\theta) = \begin{cases} C_{TP_i} & \text{if } y_i = 1 \text{ and } h_\theta(X_i) \approx 1 \\ C_{TN_i} & \text{if } y_i = 0 \text{ and } h_\theta(X_i) \approx 0 \\ C_{FP_i} & \text{if } y_i = 0 \text{ and } h_\theta(X_i) \approx 1 \\ C_{FN_i} & \text{if } y_i = 1 \text{ and } h_\theta(X_i) \approx 0 \end{cases}$$

- **Cost-Sensitive Function**

$$J^c(\theta) = \frac{1}{N} \sum_{i=1}^{N} \Big(y_i (h_\theta(X_i) C_{TP_i} + (1 - h_\theta(X_i)) C_{FN_i}) + (1 - y_i)(h_\theta(X_i) C_{FP_i} + (1 - h_\theta(X_i)) C_{TN_i}) \Big).$$

Experiments

securityandtrust.lu

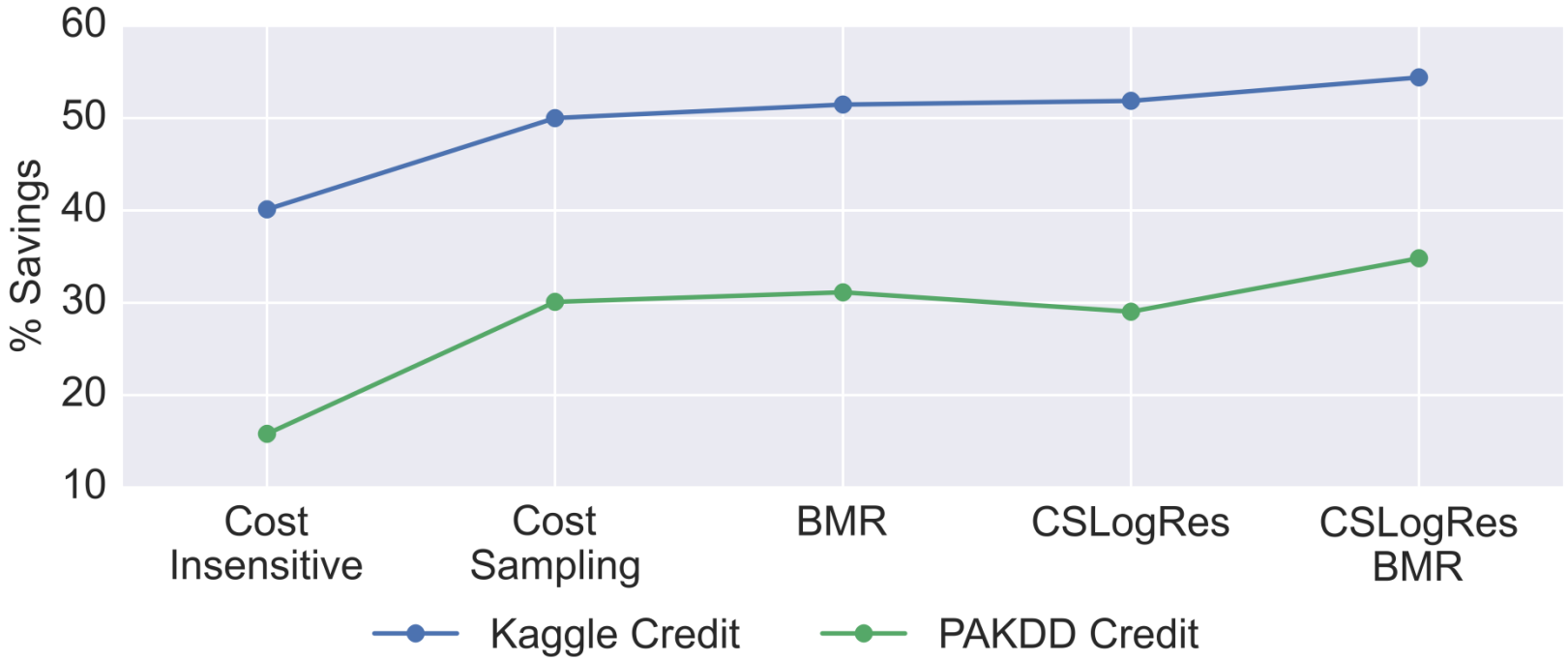
- **Savings** of the Cost-Sensitive Logistic Regression

set	Algorithm	Kaggle Credit dataset	PAKDD Credit dataset
t	$CSLR$	51.87	29.04
	$CSLR - BMR$	54.41	33.83
u	$CSLR$	-4.8	-12.17
	$CSLR - BMR$	21.65	32.46
s	$CSLR$	31.44	-6.98
	$CSLR - BMR$	34.82	32.85
r	$CSLR$	-4.8	15.91
	$CSLR - BMR$	13.16	34.54
o	$CSLR$	-4.8	20.26
	$CSLR - BMR$	38.83	34.83

UNIVERSITÉ DU LUXEMBOURG

Experiments

- Comparison of the different algorithms



Conclusion

- Selecting models based on **traditional statistics** does not give the best results in terms of cost
- Models should be evaluated taking into account **real financial costs** of the application
- Algorithms should be **developed** to incorporate those financial costs

https://github.com/albahnsen/CostSensitiveClassification

securityandtrust.lu

Thank You!!

Alejandro Correa Bahnsen

Contact information

Alejandro Correa Bahnsen

University of Luxembourg

albahnsen.com

al.bahnsen@gmail.com

http://www.linkedin.com/in/albahnsen

https://github.com/albahnsen/CostSensitiveClassification

Appendix

securitvandtrust.lu

A Calculation of a loan profit

The profit r is calculated as the present valaue of the difference between the financial institution gains and expenses, given the credit line Cl_i, the term n_i and the financial institution lending rate int_{r_i} for customer i, and the financial institution of cost funds int_{cf}.

$$r(Cl, int_r, n, int_{cf}) = PV(A(Cl, int_r, n), int_{cf}, n) - Cl, \quad (9)$$

with A being the customer monthly payment and PV the present value of the monthly payments, which are calculated using the time value of money equations [15],

$$A(Cl, int, n) = Cl\frac{int(1+int)^n}{(1+int)^n - 1}, \quad (10)$$

$$PV(a, int, n) = \frac{a}{int}\left(1 - \frac{1}{(1+int)^n}\right). \quad (11)$$

Appendix

B Calculation of the credit limit

There exist several strategies to calculate the Cl_i depending on the type of loans, the state of the economy, the current portfolio, among others [1, 15]. Nevertheless, given out lack of information regarding the specific business environment of both datasets, we simply define Cl_i as

$$Cl_i = \min\left(k \cdot Inc_i, Cl_{max}, Cl_{max}(debt_i)\right). \quad (12)$$

We fix $k = 3$ since it is the average personal loans request related to monthly income, and Cl_{max} to 25,000 Euros, which is the maximum amount for personal loans without collateral as reported by several financial institutions. Lastly, the maximum credit line given the current debt is calculated as the maximum credit limit such that the current debt ratio plus the new monthly payment does not surpass the customer monthly income. It is calculated as

$$Cl_{max}(debt_i) = PV\left(Inc_i \cdot MP_{min}(debt_i), int_r, n\right), \quad (13)$$

and

$$MP_{min}(debt_i) = \min\left(\frac{A(k \cdot Inc_i, int_r, n)}{Inc_i}, 1 - debt_i\right). \quad (14)$$